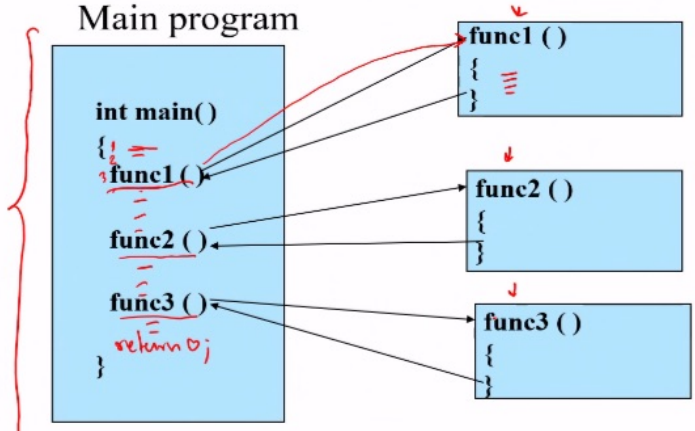
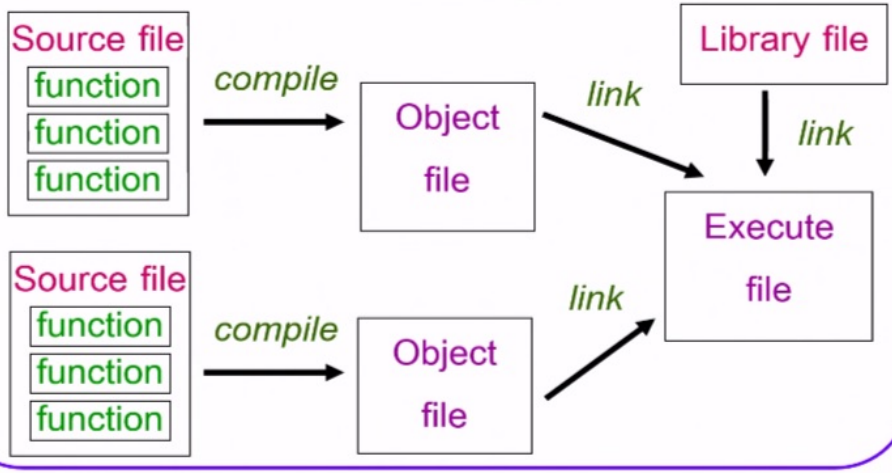
ฟังก์ชัน (Function)

การออกแบบโปรแกรมในภาษาซีจะอยู่บนพื้นฐานของการออกแบบโมดูล (Module Design) โดยการแบ่งโปรแกรมออกเป็นงานย่อย ๆ (หรือโมดูล) แต่ละงานย่อยจะทำงานอย่างใดอย่างหนึ่งเท่านั้น และไม่ควรจะมีขนาดใหญ่จนเกินไป งานย่อยเหล่านี้เมื่อนำไปเขียนโปรแกรมในภาษาซีจะเป็นการเขียนในลักษณะของฟังก์ชัน

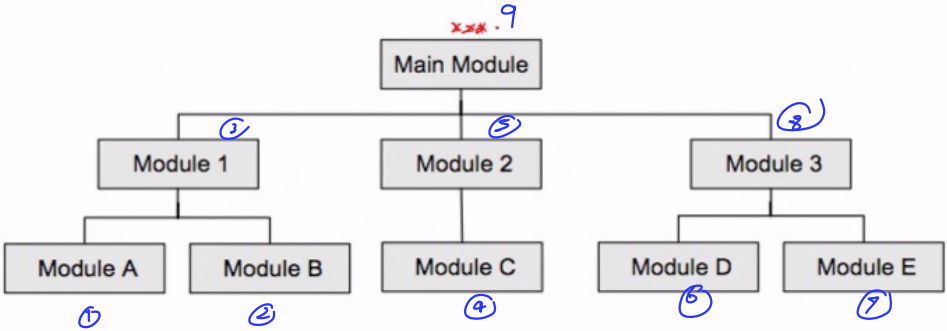
ฟังก์ชัน



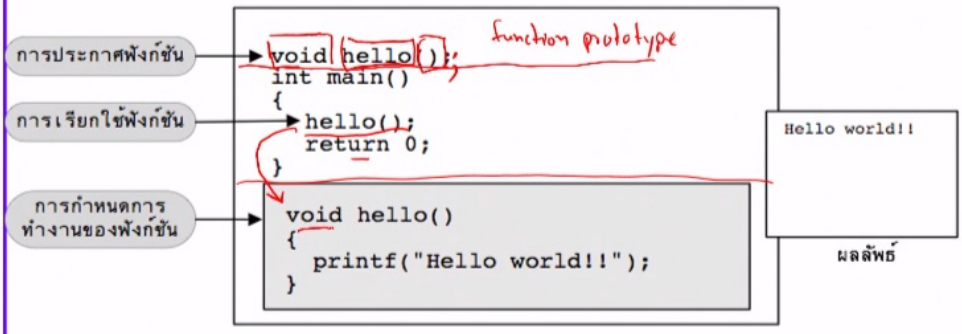
ขั้นตอนการสร้างโปรแกรมด้วยภาษา C



ฟังก์ชัน



ฟังก์ชันในภาษา C: การใช้งาน



การประกาศโพรโทไทป์ (prototype) ของฟังก์ชัน

การประกาศโพรโทไทป์เป็นสิ่งจำเป็นในภาษาซีเนื่องจากภาษาซีเป็นภาษาในลักษณะที่ต้องมีการประกาศฟังก์ชันก่อนจะเรียกใช้ฟังก์ชันนั้น **(Predefined Function)**

โพรโทไทป์ (prototype)

ส่วนประกอบหลัก 3 ส่วนของ function header

return type function_name(param 1, …,param n);

ชื่อฟังก์ชัน (function name)

พารามิเตอร์ (parameter list)

ชนิดของข้อมูลที่ส่งกลับ (function return type)

int main()
return 0;
(int)

-ออกสอบ 1 ข้อ

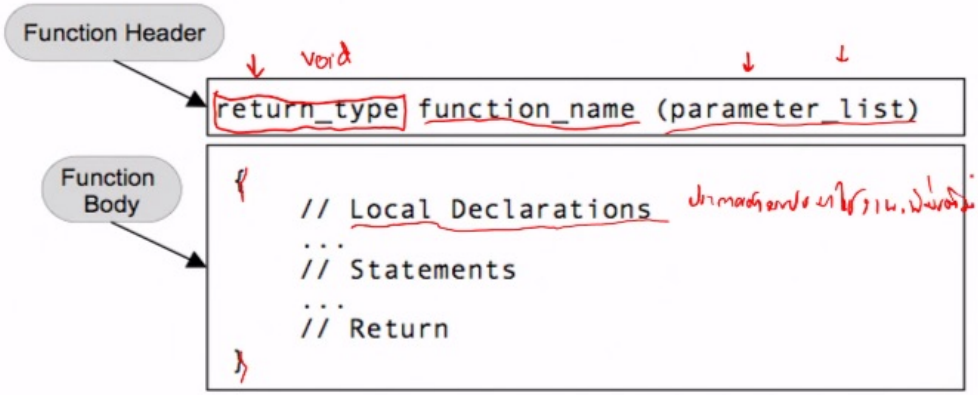
ตัวอย่าง 2 แสดงการทำงานของโปรแกรมการบวกเลขจำนวนจริง 2 จำนวนที่รับจากผู้ใช้ ในลักษณะที่มีการประกาศโปรโตไทป์

```
#include <stdio.h>
double InputDouble();
double SumDouble(double, double);
void   PrintOut(double);
```

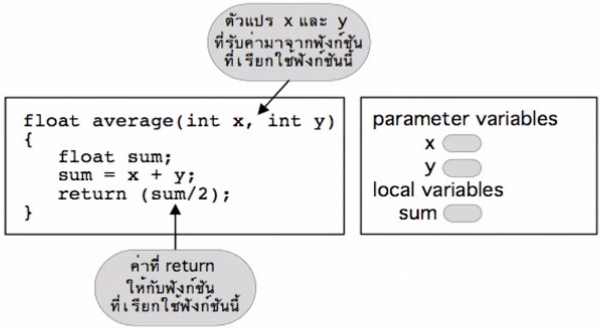

ตัวอย่าง 2 (ต่อ)

```
(int)
void main()
{
        double a1, a2, sumVal;
        a1 = InputDouble();
        a2 = InputDouble();
        sumVal = SumDouble(a1, a2);
        PrintOut(sumVal);
        return 0;
}
```

องค์ประกอบของฟังก์ชัน



Parameter และ Local variable



ตัวอย่าง

a1 และ a2 ต้องมีชนิดเป็น double
เพื่อให้ตรงกับชนิดตัวแปรของอาร์กิวเมนท์
ที่ประกาศในโปรโตไทป์

```
sumVal = SumDouble(a1,a2);
```

ใช้คู่กับโพรโทไทป์

```
double  SumDouble(double, double);
```


การออกแบบฟังก์ชัน

- วิเคราะห์การทำงานของโปรแกรม
- แบ่งการทำงานออกเป็นส่วนย่อยที่ทำงานอย่างใดอย่างหนึ่ง (งานส่วนย่อยไม่ควรมีขนาดใหญ่มาก)

ฟังก์ชันที่ไม่มีการรับค่าและส่งค่ากลับ

```
1  void hello();
2  int main()
3  {
4      printf("Function main 1\n");
5      hello();
6      printf("Function main 2\n");
7      return 0;
8  }
9  void hello()
10 {
11     printf("Hello world!!\n");
12 }
```

function Prototype

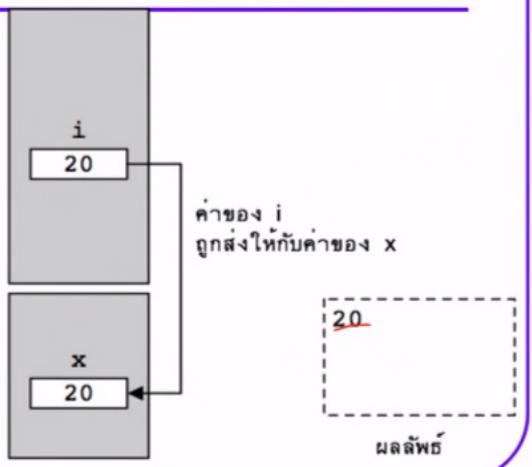
จบการทำงาน

```
Function main 1
Hello World
Function main 2
```

ผลลัพธ์

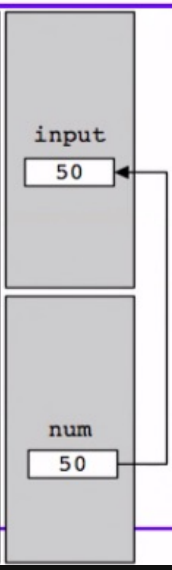
ฟังก์ชันที่มีการรับค่าแต่ไม่มีการส่งค่ากลับ

```
1  void showValue(int x);
2  int main()
3  {
4      int i = 20;
5      showValue(i);
6      return 0;
7  }
8  void showValue(int x)
9  {
10     printf("%d\n",x);
11 }
```



ฟังก์ชันที่ไม่มีการรับค่าแต่มีการส่งค่ากลับ

```
1  int getInput();
2  int main()
3  {
4      int input;
5      input = getInput();
6      printf("input = %d", input);
7      return 0;
8  }
9  int getInput()
10 {
11     int num;
12     printf("Input number");
13     scanf("%d",&num);
14     return num;
15 }
```



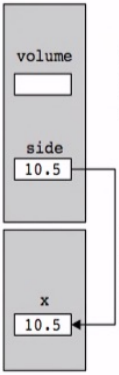
ค่าที่ return ผ่านตัวแปร num จากฟังก์ชัน getInput ถูกกำหนดค่าให้ตัวแปร input ในฟังก์ชัน main

```
Input number: 50
input = 50
```

ผลลัพธ์ สมมติว่ารับค่า 50

ฟังก์ชันที่มีการรับค่าแต่มีการส่งค่ากลับ

```
1  float cubeVol(float x);
2  int main()
3  {
4      float side = 10.5;
5      float volume;
6      volume = cubeVol(side);
7      printf("Volume of cube = %f",volume);
8      return 0;
9  }
10 float cubeVol(float x)
11 {
12     return x * x * x;
13 }
```



ค่าที่ return จากฟังก์ชัน cubeVol ถูกกำหนดให้ ตัวแปร volume

```
Volume of cube = 1157.625
```

ผลลัพธ์

การส่งพารามิเตอร์ให้กับฟังก์ชัน
Argument & Parameter

```
void printValue (int x, float y);
int main()
{
    int a = 10;
    ...
    printValue (a, 20.5);
    ...
}

void printValue (int x, float y)
{
     printf("%d %f \n", x, y);
}
```

argument ที่ส่งให้ฟังก์ชัน printValue

int float

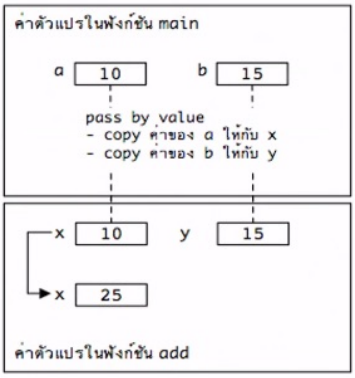
- จำนวนของ argument ต้องเท่ากับจำนวนของ parameter
- ชนิดข้อมูลของ argument ต้องเหมือนกับชนิดข้อมูลของ parameter

parameter ของฟังก์ชัน printValue

Pass by value

```
void add(int x, int y);
int main()
{
  int a = 10;
  int b = 15;
  printf("a = %d\n",a);
  add(a,b);
  printf("a = %d\n",a);
  return 0;
}
void add(int x, int y)
{
  x = x + y;
  printf("x = %d\n", x);
}
```

ค่าของ x เปลี่ยนเป็น 25



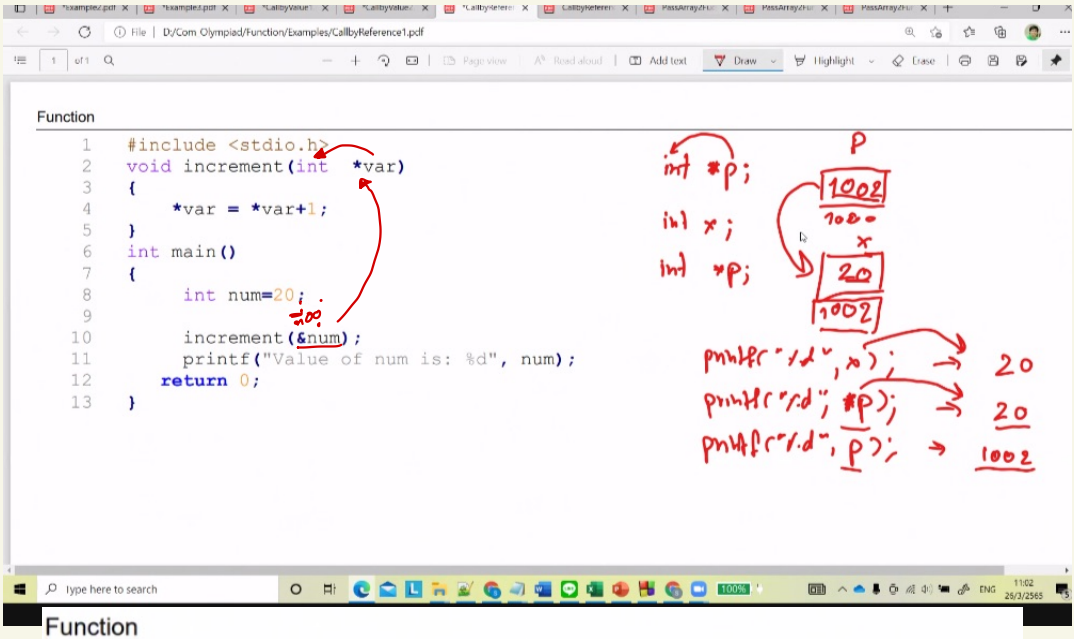
Global Variable

นอกจากนี้สามารถประกาศตัวแปรไว้ที่ภายนอกฟังก์ชัน บริเวณส่วนเริ่มของโปรแกรมจะเรียกว่า

ตัวแปรโกลบอล (Global Variable) ซึ่งเป็นตัวแปรที่สามารถเรียกใช้ที่ตำแหน่งใด ๆ ในโปรแกรมก็ได้ ยกเว้นในกรณีที่มีการประกาศตัวแปรที่มีชื่อเดียวกันตัวแปรโกลบอลภายในบล็อกหรือฟังก์ชัน

ข้อควรระวังในการใช้งานฟังก์ชัน

- ต้องมีการประกาศฟังก์ชันก่อนการใช้งานฟังก์ชัน การส่งผ่านค่า **`argument`** ส่งให้ถูกต้องและห้ามส่งสลับที่กัน จะทำให้การทำงานผิดได้



Function

```c
#include
void swapnum ( int *var1, int *var2 )
{
    int tempnum ;
    tempnum = *var1 ;
    *var1 = *var2 ;
    *var2 = tempnum ;
}
int main( )
{
    int num1 = 35, num2 = 45 ;
    printf("Before swapping:");
    printf("\nnum1 value is %d", num1);
    printf("\nnum2 value is %d", num2);

    /*calling swap function*/
    swapnum( &num1, &num2 );

    printf("\nAfter swapping:");
    printf("\nnum1 value is %d", num1);
    printf("\nnum2 value is %d", num2);
    return 0;
}
```

5.

โปรแกรมต่อไปนี้มีผลลัพธ์แสดงทางจอภาพอย่างไร

```
1  #include<stdio.h>
2  void function(int *x, *in y);
3  int main()
4  {
5      int x = 13;
6      int y = 6;
7      printf("x = %d, y = %d \n", x, y);
8      function(&x, &y);
9      printf("x = %d, y = %d \n", x, y);
10     return 0;
11 }
12
13 void function(int *x, int *y)
14 {
15     int temp;
16     if(*x > *y){
17         temp = *x;
18         *x = *y;
19         *y = temp * 2;
20     }
21 }
```

คำตอบของคุณ

File | D:/Com_Olympiad/Function/Examples/PassArray2Fucntion1.pdf

Function

```c
#include <stdio.h>
void disp( char ch)
{
   printf("%c ", ch);
}

int main()
{
   char arr[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'};
   for (int x=0; x<10; x++)
   {
       /* I'm passing each element one by one using subscript*/
       disp (arr[x]);
   }

   return 0;
}
```

Type here to search

File | D:/Com_Olympiad/Function/Examples/PassArray2Function2.pdf

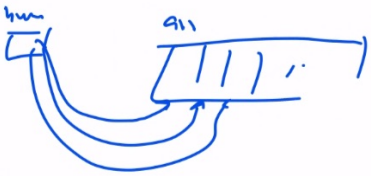
Function

pass by reference

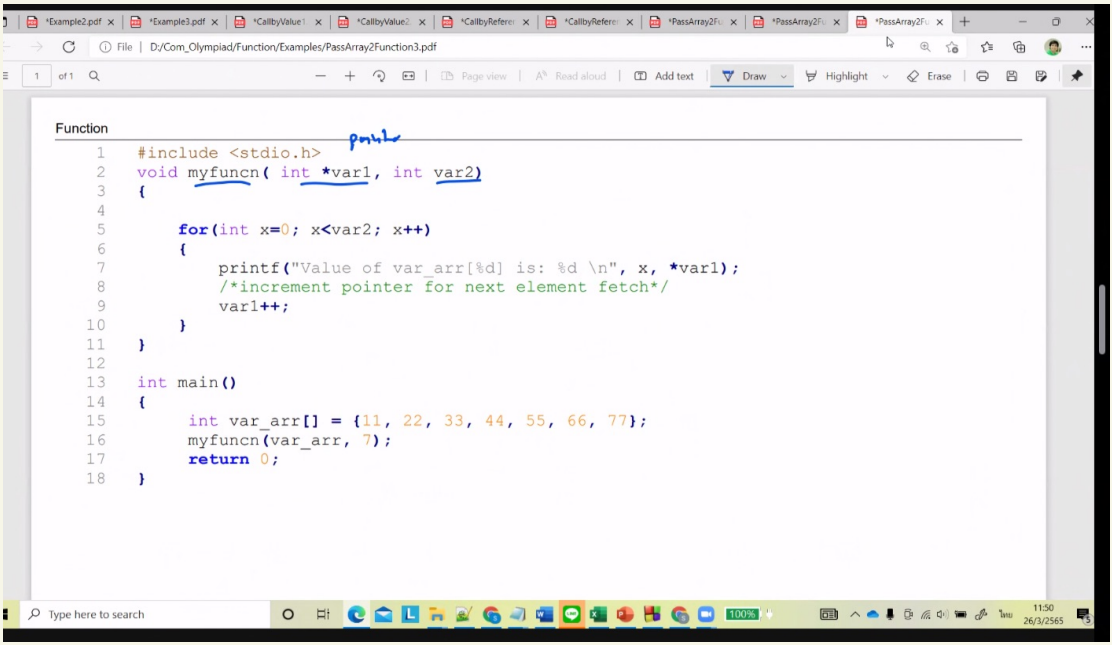
```c
#include <stdio.h>
void disp( int *num)
{
    printf("%d ", *num);
}

int main()
{
     int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
     for (int i=0; i<10; i++)
     {
         /* Passing addresses of array elements*/
         disp (&arr[i]);
     }

     return 0;
}
```



Type here to search


*Example2.pdf
*Example3.pdf
*CallbyValue1
*CallbyValue2
*CallbyRefere
*CallbyRefere
*PassArray2Fu
*PassArray2Fu
*PassArray2Fu
File | D:/Com_Olympiad/Function/Examples/PassArray2Function3.pdf
1 of 1
Page view
Read aloud
Add text
Draw
Highlight
Erase
Function
pass
#include <stdio.h>
void myfuncn(int *var1, int var2)
{
for(int x=0; x<var2; x++)
{
printf("Value of var_arr[%d] is: %d \n", x, *var1);
/*increment pointer for next element fetch*/
var1++;
}
}
int main()
{
int var_arr[] = {11, 22, 33, 44, 55, 66, 77};
myfuncn(var_arr, 7);
return 0;
}
Type here to search
100%
11:50
26/3/2565

#include<math.h>

ตัวอย่างฟังก์ชัน	ความหมาย	ชนิดของตัวแปรเข้า	ชนิดของผลลัพธ์
abs(x)	ค่าสัมบูรณ์ของจำนวนเต็ม x	int	int
labs(x)	ค่าสัมบูรณ์ของจำนวนเต็มความจุสูง x	long	long
fabs(x)	ค่าสัมบูรณ์ของจำนวนจริง x	double	double
tan(x)	ค่า tangent ของ x	double	double
sin(x)	ค่า sine ของ x	double	double
cos(x)	ค่า cosine ของ x	double	double
pow(x, y)	ค่า x ยกกำลัง y	double	double
fmod(x, y)	ค่าเศษจากการหาร x/y แบบจำนวนจริง	double	double

ตัวอย่าง

```
#include<stdio.h>
#include<math.h>
#define PI 3.141592654
int main()
{
        double degree;
        double radian;
        degree = 60.0;
        radian = degree * PI/180;
        printf("degree = %f\n", degree);
        printf("radian  = %f\n", radian);
        printf("cos     = %f\n", cos(radian));
        printf("sin     = %f\n", sin(radian));
        printf("tan     = %f\n", tan(radian));
        getch();
        return 0;
}
```

ตัวอย่าง

```
pow     = 9.0000
sqrt    = 4.0000
exp     = 7.3891
log_e   = 2.7726
Log_10  = 1.2041
```

```
#include<stdio.h>
#include<math.h>

int main()
{
        double a = 3.0;
        double b = 2.0;
        double c = 16.0;

        printf("%.4f\n", pow(a,b));
        printf("%.4f\n", sqrt(c));
        printf("%.4f\n", exp(b));
        printf("%.4f\n", log(c));
        printf("%.4f\n", log10(c));

        getch();
        return 0;
}
```


#include<math.h>

ตัวอย่างฟังก์ชัน	ความหมาย	ชนิดของตัวแปรเข้า	ชนิดของผลลัพธ์
exp(x)	คำนวณค่า e ของ x	double	double
log(x)	คำนวณค่า log ฐาน e ของ x	double	double
log10(x)	คำนวณค่า log ฐาน 10 ของ x	double	double
atof(x)	เปลี่ยนกลุ่มอักขระ x ให้เป็นจำนวนจริง	char	double
atoi(x)	เปลี่ยนกลุ่มอักขระ x ให้เป็นจำนวนเต็ม	char	integer
atol(x)	เปลี่ยนกลุ่มอักขระ x ให้เป็นจำนวนเต็มความถูสูง	char	long
ceil(x)	ค่าจำนวนเต็มที่น้อยที่สุดที่มากกว่าหรือเท่ากับ	double	double
floor(x)	ค่าจำนวนเต็มที่มากที่สุดที่น้อยกว่าหรือเท่ากับ x	double	double
sqrt(x)	ค่ารากที่สองของ x	double	double